

AMENDMENTS TO THE CLAIMS:

This listing of claims will replace all prior versions and listings of claims in the application:

1. (Currently Amended) A method of processing data comprising:

defining an instance of a class, the class ~~supporting~~ having

defined fields, each of the defined fields having a field value, a field name, and a field data type, wherein memory space for each field value is allocated when the instance of the class is created, and

defined options, each of the defined options having an option value, an option name, and an option data type, wherein memory space for the option value is allocated when each of the options corresponding to the option value is optionally associated with ~~the options are referenceable without allocation of memory space for the option values when the instance of the class is created;~~ accessing field values and option values in the instance using program code expressions having the same syntactic form, each of the program code expressions comprising an operator and a property name, the accessing further comprising

accessing a selected field value in the instance using a first single program code expression, the property name in the first single program code expression ~~comprising an operator and the~~ a field name corresponding to the selected field value, and

accessing a selected option value in the instance using a second single program code expression, the property name in the second single program code

expression comprising ~~the operator and the~~ an option name corresponding to the selected option value; and

during compilation, determining whether at least one of the program code expressions accesses one of (a) the selected field value or (b) the selected option value based on the property name in the at least one of the program code expressions;

when it is determined that the selected field value is being accessed, compiling the first ~~single~~ program code expression into a first code for accessing the selected field value; and

when it is determined that the selected option value is being accessed, compiling the second ~~single~~ program code expression into a second code for accessing the selected option value.

2. (Previously Presented) A method as claimed in claim 1 wherein a selected option is associated with change handler code that is executed when the option value corresponding to the selected option changes.

3. (Currently Amended) A method as claimed in claim 2 wherein the change handler code associated with the selected option is defined in a plurality of classes within a class inheritance hierarchy and the change handler code from each of the plurality of classes is executed when the option value corresponding to the selected option changes.

4. (Previously Presented) A method as claimed in claim 1, further comprising:

during compilation of an operation on the selected option value, using the corresponding option data type to process the operation.

5. (Currently Amended) A method as claimed in claim 1 wherein a selected option is associated with a default value, the method further comprising, in a operation to the selected option in the instance of the class, if the option value corresponding to the [[the]] selected option has been set, getting the set option value and, if the option value corresponding to the selected option has not been set, getting the associated default value.

6. (Currently Amended) A method as claimed in claim 1 wherein defining [[an]] the instance comprises:

defining a first instance of a first class, the first class supporting a first set of options, wherein the first set of options is associated with a first listing data structure of a first form, the method further comprising:

defining a second instance of a second class, the second class supporting a second set of options without allocation of memory space for [[the]] option values corresponding to the second set of options when the second instance is created, wherein the second set of options is associated with a second listing data structure of a second form, the second form being different from the first form;

during compilation, encoding an operation on the first instance as a method call to the first instance of the first class without regard to form of the first listing data structure or the second listing data structure; and

during compilation, encoding an operation on the second instance without regard to form of the first listing data structure or the second listing data structure.

7. (Previously Presented) A method as claimed in claim 1 further comprising:

notifying the instance of the class of a change in the selected option value through a change handler identified by an option binding object, the option binding object being located by first searching a mapping data structure for any mapping from the option name corresponding to the selected option value to the option binding object and, if no mapping was found, by then computing a mapping from the option name corresponding to the selected option value to the option binding object and storing the mapping in the mapping data structure.

8. (Currently Amended) A method as claimed in claim 1 further comprising providing referenced options comprising options in the instance that have been referenced, wherein the instance of the class is associated with a listing data structure, the listing data structure comprising option items,

each of the option items corresponding to a referenced option ~~comprising an option in the instance that has been referenced~~,

each of the option items having the option value and the option name corresponding to one of the referenced options, and

the option items being arranged in a linked list;

wherein the method further comprises, when a first option is referenced in order to set a first option value for the first option, checking the listing data structure for a first option item corresponding to the first option;

when the first option item is found, setting the first option value in the first option item; and

when no first option item is found, creating the first option item, setting the first option value in the first option item, and storing the first option item with the set first option value in the listing data structure.

9. (Previously Presented) A method as claimed in claim 1 wherein a nonlocal option value applies to other instances of the class in a nonlocal option hierarchy.

10. (Original) A method as claimed in claim 9 wherein the nonlocal option hierarchy is a graphical hierarchy.

11. (Currently Amended) A data processing system, comprising:
a processor for processing an instance ~~instances~~ of a class, the ~~instances~~
instance of the class comprising:

defined fields, each of the defined fields having a field value, a field name, and a field data type, wherein memory space for each field value is allocated when ~~[[an]]~~ the instance of the class is created; and

defined options, each of the defined options having an option value, an option name, and an option data type, wherein memory space for the option value is allocated when each of the options corresponding to the option value is optionally associated with ~~the options are referenceable without allocation of memory space for the option value when~~ the instance of the class is created; ~~the data processing system further comprising:~~

the processor configured to access field values and option values in the instance using program code expressions having the same syntactic form, each of the program code expressions comprising an operator and a property name, the processor configured to implement program code

wherein a selected field value in the instance is accessed using a first single program code expression, the property name in the first single program code expression comprising ~~an operator and the~~ a field name corresponding to the selected field value, and

wherein a selected option value in the instance is accessed using a second single program code expression, the property name in the second single program code expression comprising ~~the operator and the~~ an option name corresponding to the selected option value;

memory having the memory space that is allocable for the option value when the each of the options corresponding to the option value is optionally associated with the instance of the class; and

a compiler which determines whether at least one of the program code expressions accesses one of (a) the selected field value or (b) the selected option value based on the property name in the at least one of the program code expressions, wherein,

when the compiler determines that the selected field value is being accessed, the compiler compiles the first ~~single~~ program code expression into a first code for accessing the selected field value; and

when the compiler determines that the selected option value is being accessed, the compiler compiles the second ~~single~~ program code expression into a second code for accessing the selected option value.

12. (Previously Presented) A system as claimed in claim 11 wherein a selected option is associated with change handler code that is executed when the option value corresponding to the selected option changes.

13. (Currently Amended) A system as claimed in claim 12 wherein the change handler code associated with the selected option is defined in a plurality of classes within a class inheritance hierarchy and the change handler code from each of the plurality of classes is executed when the option value corresponding to the selected option changes.

14. (Currently Amended) A system as claimed in claim 11, wherein the further comprising a compiler is further configured which, during compilation of an operation on the selected option value, [[uses]] to use the corresponding option data type to process the operation.

15. (Previously Presented) A system as claimed in claim 11 wherein a selected option is associated with a default value that is obtained when no value has been set for the selected option.

16. (Currently Amended) A system as claimed in claim 11 comprising a plurality of classes supporting options, wherein the options supported by each class of the classes are associated with listing data structures having different forms, and wherein the compiler ~~encodes~~ is configured to encode an operation on instances of the classes as a method call to [[an]] selected instance of one of the classes without regard to form of the listing data structure associated therewith.

17. (Currently Amended) A system as claimed in claim 11 further comprising change handlers which notify instances of the class of a change in a supported option value, and

a mapping data structure which maps a supported option name corresponding to the supported option value ~~and class~~ to an option binding object, wherein the option binding object identifies a change handler corresponding to the supported option value.

18. (Currently Amended) A system as claimed in claim 11,
wherein the instance of the class is associated with a listing data structure
comprising option items,
each of the option items corresponding to a referenced option comprising
an option that has been referenced in the instance,
each of the option items having the option value and the option name
corresponding to one of the referenced options, and
the option items being arranged in a linked list;
wherein, when a first option is referenced in order to set a first option value for
the first option, the processor is configured to check the linked list ~~is checked~~ for a first
option item corresponding to the first option,
when the first option item is found, the processor is configured to set the
first option value ~~is set~~ in the first option item; and
when no first option item corresponding to the first option is found, the
processor is configured to create the first option item ~~is created~~, to set the first
option value ~~is set~~ in the first option item, and to store the first option item with
the set first option value ~~is stored~~ in the listing data structure.

19. (Previously Presented) A system as claimed in claim 11 wherein a
nonlocal option value applies to other instances of the class in a nonlocal option
hierarchy.

20. (Original) A system as claimed in claim 19 wherein the nonlocal option hierarchy is a graphical hierarchy.

21. (Currently Amended) A data processing system comprising:
means for defining an instance of a class, the class ~~supporting~~ having
defined fields, each of the defined fields having a field value, a field name,
and a field data type, wherein memory space for each field value is allocated
when the instance of the class is created, and
defined options, each of the defined options having an option value, an
option name, and an option data type, wherein memory space for the option
value is allocated when each of the options corresponding to the option value is
optionally associated with ~~the options are referenceable without allocation of~~
~~memory space for the option values when the instance of the class is created;~~
means for accessing field values and option values in the instance using program
code expressions having the same syntactic form, each of the program code
expressions comprising an operator and a property name, the means for accessing
further comprising
means for accessing a selected field value in the instance using a first
single program code expression, the property name in the first single program
code expression comprising ~~an operator and the~~ a field name corresponding to
the selected field value, and means for accessing a selected option value in the
instance using a second single program code expression, the property name in

the second single program code expression comprising the operator and the an
option name corresponding to the selected option value; and
means for allocating memory space for the option value when the each of the
options corresponding to the option value is optionally associated with the instance of
the class;

means for determining, during compilation, whether at least one of the program
code expressions accesses one of (a) the selected field value or (b) the selected option
value based on the property name in the at least one of the program code expressions;

when it is determined that the selected field value is being accessed,
compiling the first single program code expression into a first code for accessing
the selected field value; and

when it is determined that the selected option value is being accessed,
compiling the second single program code expression into a second code for
accessing the selected option value.

22. (Currently Amended) A computer program product comprising:
a computer usable medium for storing data; and
a set of computer program instructions embodied on the computer usable
medium, including instructions for:

defining an instance of a class, the class supporting having

defined fields, each of the defined fields having a field value, a field name,
and a field data type, wherein memory space for each field value is allocated
when the instance of the class is created and

defined options, each of the defined options having an option value, an option name, and an option data type, wherein memory space for the option value is allocated when each of the options corresponding to the option value is optionally associated with ~~the options are referenceable without allocation of memory space for the option values when the instance is created;~~ accessing field values and option values in the instance using program code expressions having the same syntactic form, each of the program code expressions comprising an operator and a property name, the accessing further comprising

accessing a selected field value in the instance using a first single program code expression, the property name for the first single program code expression comprising ~~an operator and the~~ a field name corresponding to the selected option value, and

accessing a selected option value in the instance using a second single program code expression, the property name for the second single program code expression comprising ~~the operator and the~~ an option name corresponding to the selected option value; and

during compilation, determining whether at least one of the program code expressions accesses one of (a) the selected field value or (b) the selected option value based on the property name in the at least one of the program code expressions;

when it is determined that the selected field value is being accessed, compiling the first single program code expression into a first code for accessing the selected field value; and

when it is determined that the selected option value is being accessed, compiling the second ~~single~~ program code expression into a second code for accessing the selected option value.

23. (Previously Presented) A product as claimed in claim 22 wherein the computer program instructions include instructions to notify instances of the class of a change in a supported option value.

24. (Currently Amended) A product as claimed in claim 22, further comprising referenced options comprising options in the instance that have been referenced.

wherein the defined options are associated with a listing data structure ~~comprises~~ comprising a linked list of option items,

each of the option items corresponding to a referenced option ~~an option that has been referenced in the instance,~~ and

each of the option items having the option value and the option name corresponding to one of the referenced options; and

wherein the linked list is checked when a first option is referenced in order to set a first option value for the first option, and

when the first option item is found, the first option value is set in the first option item; and

when no first option item corresponding to the first option is found, the first option item corresponding to the first option is created, the first option value is set in the first

option item, and the first option item with the set first option value is stored in the linked list.

25-30. (Cancelled).